

Apprendre l'informatique par la programmation des robots

Claver Nijimbere, Lætitia Boulc'h, Mariam Haspekian, Georges Louis Baron

njmbrcivr@yahoo.fr, laetitia.boulch@laposte.net ; mariam.haspekian@parisdescartes.fr,

georges-louis.baron@parisdescartes.fr

Université Paris Descartes, Laboratoire EDA

Résumé. Le texte qui suit vise la construction des savoirs informatiques chez les débutants et s'inscrit dans le cadre d'une thèse de doctorat en cours à l'université Paris Descartes. Il s'intéresse à un aspect particulier de l'apprentissage de l'informatique – la programmation des robots – chez les étudiants de licence en début d'une spécialisation informatique. Des observations ethnographiques des pratiques en contexte de projets et des entretiens périodiques en groupes, ont été effectués, puis complétés par des entretiens en groupe conduits en fin de projets auprès des enseignants-encadrants. Toutes ces données ont été analysées et discutées à la lumière de la littérature disponible. Les résultats montrent beaucoup de connaissances informatiques construites et une grande utilisation des savoirs pluridisciplinaires déjà acquis.

Mots-clés : Robotique MINDSTORMS NXT, didactique, pédagogie par projets, apprentissage, programmation informatique

Contexte : les projets robotiques, des solutions possibles aux problèmes de l'apprentissage de l'informatique en licence

L'apprentissage de la plupart des disciplines scientifiques souffre d'un manque de motivation chez les jeunes. Un sentiment d'inaccessibilité voire de rejet a d'ailleurs été développé chez beaucoup parmi eux et le caractère abstrait des sciences en serait à l'origine (Nonnon, 2002, p.33). C'est aussi le cas pour l'informatique. Janiszek, Pellier, Mauclair, Boulc'h, Baron & Parchemal (2011a, p.1) indiquent que ce manque de motivation de l'informatique chez les jeunes a poussé à imaginer et élaborer des parcours pédagogiques plus attractifs et plus adaptés aux étudiants pour l'apprentissage motivé de l'informatique. Si l'intérêt de la robotique pédagogique est de plus en plus évoqué actuellement dans l'enseignement de l'informatique, cette question n'est pas nouvelle : elle a fait objet de recherches vers le début des années 90 (Baron et Denis, 1993 ; Duchâteau, 1993). Dressant un parallélisme entre la programmation informatique classique et la robotique, Charles Duchâteau (1993, p.1) révèle une procédure commune de la démarche conceptuelle de ces deux domaines informatique, ce qui, par conséquent, peut leur permettre d'être analysées sous la même grille théorique. La robotique pédagogique, l'une des récentes innovations technologiques et particulièrement le robot humanoïde, est vue comme un outil d'apprentissage idéal pour les classes de tous les niveaux¹. Actuellement, l'approche orientée projet de la programmation des robots est privilégiée dans l'apprentissage de l'informatique. Cette approche est venue renforcer l'idée de Charles Duchâteau qui, il y a une vingtaine d'années (Duchâteau, 1993, p.11), soulignait l'intérêt de la robotique pédagogique dans l'apprentissage de l'informatique et recommandait de « consacrer un certain temps à la programmation de robots » au cours de l'apprentissage de la programmation. Encouragée dans plusieurs domaines dont le milieu éducatif (Bonnell, 2010), l'utilisation de la robotique pour les apprentissages scolaires semble prendre de plus en plus de l'ampleur, même si sa cherté fait qu'elle reste encore le monopole des établissements privilégiés (Janiszek et al., 2011a, p.3). Dans leur article récent, Gaudiello & Zibetti (2013, p.18) parlent de révolution technologique marquant ce début du XXI^e siècle et conduisant à l'« ère du robot ». Selon eux, l'intérêt estimé pour ce genre de technologie – le robot – a

1 <http://www.aldebaran-robotics.com/For-Education/introduction.html>, consulté le 20 septembre 2012

permis, en éducation, l'émergence d'un nouveau champ d'étude spécifique : la robotique éducative. Le choix de ce type d'approche par projet robotique est justifié par la motivation des apprenants les rendant actifs dans la résolution des problèmes prescrits au cours de laquelle les conséquences de leurs décisions sont clairement et immédiatement visibles (Janiszek et al, 2011a, p.1).

Pour Arnaud (1999), cette approche pédagogique, vue comme stimulante pour des étudiants déjà intéressés par cette discipline, entraîne un effet positif sur l'assimilation des connaissances. Il rejoint ainsi le point de vue de Tardif (1997) qui évoquait, deux ans avant lui, la recherche de solutions mises en œuvre au cours de l'activité. Pour Tardif, cette assimilation est rendue possible par le fait que « ils (les étudiants) ne vont pas seulement appliquer une série de procédures et de contenus mémorisés mais procéder à une recherche de solution en prenant appui sur les acquis théoriques qui seront réinvestis dans cette situation problème ». Cette liaison théorie-pratique permet ainsi de construire un ensemble de connaissances issues de beaucoup de domaines : informatique, ingénierie, mathématiques, la physique²... La robotique pédagogique est donc inscrite dans un contexte de l'approche constructiviste de l'apprentissage en utilisant un ordinateur connecté au monde physique (Nonnon, 2002, p.40). Selon cet auteur, elle est une alternative aux cours magistraux, aux activités de laboratoires traditionnels et ajoute une option supplémentaire aux programmes de simulation. D'un point de vue didactique, elle est, selon lui, un dispositif technologique supposant un processus pédagogique de résolution systématique des problèmes et, permettant de capitaliser sur le goût des apprenants pour le concret. Par son appartenance au champ de la didactique, les manipulations concrètes qu'elle permet aux étudiants les amènent à s'approprier des « gestes mentaux » qui font partie intégrante des processus scientifique et la construction des opérations mentales (Nonnon, 2002, p. 40)

Nous nous sommes intéressés aux étudiants de licence (L2), en informatique à l'université Paris Descartes. Si notre sujet de thèse s'intéresse à l'apprentissage de l'informatique chez les débutants, il est estimé que, dans ces conditions, ces étudiants ne sont pas loin des débutants en informatique. Ils ont suivi une année commune en première année de licence (L1) de mathématiques et informatique avec une moitié de modules mathématiques et une autre moitié de modules informatiques. Nous nous sommes en particulier interrogé sur les questions suivantes : comment les étudiants de licence apprennent-ils lors de la programmation des robots ? Quelles connaissances construisent-ils au cours de ces projets ? Quels sont les objectifs poursuivis par les projets de programmation en licence ? Quelles sont les connaissances construites au cours des projets de programmation des robots ? Quelles les méthodes ou stratégies sont mises en œuvre au cours de cette activité ? Quel est l'apport de l'approche par projet dans la construction des connaissances chez des étudiants ?

Méthodologie

Brève présentation du terrain de jeu et du robot utilisés

Nous avons observé la construction de projets consistant à programmer un jeu où deux robots adverses visent à envoyer des palets dans les buts adverses. Entouré d'une bordure rigide de 15 cm, le terrain de jeu mesure 3 mètres sur 2. Les zones de terrain sont délimitées par des lignes de couleurs différentes : gris clair pour le fond du terrain, blanc pour la limite des buts, vert et bleu pour les lignes Est et Ouest du terrain respectivement, rouge et jaune pour le Nord et le Sud respectivement et, noire pour les lignes partageant le terrain en son milieu de l'Est à l'Ouest et du Nord au Sud. Des palets³, au nombre de neuf, ont été positionnés aux

2 <http://www.aldebaran-robotics.com/For-Education/introduction.html>, consulté le 20 septembre 2012

3 Alors qu'initialement il était prévu d'utiliser des balles, les organisateurs ont remplacé par après ces balles

intersections de ces lignes (Règlement de la compétition, 2012).

Le robot LEGO MINDSTORM NXT utilisé (fig.1), que nous appellerons simplement robot dans la suite, est un kit MINDSTORMS NXT assemblable et programmable. Fabriqué par la société LEGO⁴, il est principalement formé d'une « brique intelligente » NXT – son « cerveau » -, programmée en langage NXC, proche du langage C. Cette brique se caractérise par un microprocesseur 32 bits, de fonction bluetooth, de ports d'entrée et de sortie, de haut-parleur intégré, etc. Peu évolué, il dispose d'actionneurs et de capteurs de pression, de couleur, de sons et d'ultrasons⁵, des servomoteurs pour les capteurs de mouvement, de rotations, mais avec une résistance mouvement et ses deux pinces lui permettent de ramasser des palets⁶. Avec tous ces éléments, il est capable de s'approprier des informations de son milieu et peut facilement s'y déplacer. Il est, en effet, muni de trois roues dont l'arrière mobile librement.



Figure 1 : Le robot LEGO MINDSTORM NXT

Des recherches se sont intéressées aux différents types de technologies robotiques. Elles ont relevé beaucoup de potentialités éducatives. Les kits robotique dont fait partie LEGO MINDSTORM NXT, renferment de vastes possibilités de construction et de programmation, offertes par le caractère à la fois interactif et transparent de ce type de technologie (Kynigos, 2008, p.1). Cette transparence des kits robotiques est définie par opposition aux technologies humanoïdes dont NAO qualifiées de « black box », - non transparence -, donc hermétique à la programmation de ses comportements par les élèves ou étudiants. Selon lui, une telle combinaison semble favoriser une approche éducative au profit de l'apprenant : elle lui donne l'occasion de devenir auteur plutôt que consommateur de la technologie.

Collecte de données : une approche qualitative

La recherche a concerné treize (13) étudiants, répartis en trois groupes : un groupe de cinq noté L2_DJ_5H⁷ et deux groupes de quatre chacun, notés L2_DJ_4H et L2_MS_2H. Malgré l'existence de quelques rares exceptions qui ont suivi des filières informatiques au lycée, beaucoup parmi eux n'ont été en contact avec des cours d'informatique qu'à l'université. Ils ont tous travaillé sur un même sujet de projet : la programmation du robot, « ramasseur de balles ». La recherche a aussi et deux (2) enseignants. La population a compris aussi deux

par des palets, faciles à ramasser pour le robot

4 <http://mindstorms.lego.com/en-us/default.aspx>

5 Les capteurs à ultrason permettent de détecter des objets et de mesurer les distances

6 Les palets ont été utilisés à la place des balles pour rendre facile la tâche de ramassage : palets tiennent bien dans les pinces de LEGO

7 Dans la notation des groupes, L2 représente la deuxième année de licence, le binôme suivant représente le nom et prénom de l'encadrant et le binôme suivant le numéro du projet sur lequel le groupe travail.

enseignants, notés E1 et E2 dans la suite.

Une méthode qualitative de type ethnographique a été privilégiée pour répondre à ce questionnement. Jean François Marcel (2002) justifie la pertinence de ce type d'approche par quatre raisons principales : la nécessité d'une observation directe et prolongée, la seule à même de permettre d'accéder à la description d'un peu connu dans ses éventuelles spécificités ; la volonté de combiner des éléments facteurs et observables avec le sens que leur donnent les acteurs, une situation permettant de qualifier le dispositif d'« observation participante » ; le recours à la technique de notes de terrain rédigées « à chaud » juste après l'observation ; la méthodologie inductive de traitement des données correspondant à la visée exploratoire et enfin, la théorisation des matériaux empiriques par laquelle elle permet d'accéder par la connaissance des pratiques, constitue une étape propédeutique à la stabilisation de l'instrumentation.

Dans cette approche qualitative, des observations de pratiques ont été faites pendant tout le second semestre de l'année scolaire 2011/2012. Deux moments ont été particulièrement privilégiés dans nos observations : lors du travail en salle de robotique et lors des rencontres entre un encadrant et un groupe donné d'étudiants. En salle de robotique, l'ancrage des observations était l'organisation des groupes dans leurs apprentissages et aux interactions entre eux : entre les membres d'un groupe, entre les groupes, entre étudiants et le système ordinateur-robot, les stratégies utilisées par les différents groupes d'étudiants dans la programmation des tâches, les difficultés et contraintes vécues et les connaissances utilisées d'une part et celles construites d'autre part. En réunion entre étudiants-encadrant, les observations s'intéressaient davantage aux types de problèmes soumis par les étudiants au type d'aide donnée par l'encadrant. Elles ont été complétées par des entretiens semi directifs, d'abord avec les étudiants puis avec les enseignants-encadrants. Pour chaque groupe d'étudiants, plusieurs entretiens ont été menés : au début, au milieu et à la fin des projets. Les premiers ont eu lieu, en janvier. Ils étaient relatifs aux cahiers de charges afin de répondre à plusieurs questions : comment les étudiants ont-ils choisi leur sujet de projet et quelles sont les motivations de leurs choix ; quel était, selon eux, l'intérêt de leurs projets mais aussi pourquoi les enseignants (du département d'informatique à l'Université Paris Descartes), avaient jugé intéressant de les leur proposer. Enfin, ces entretiens initiaux cherchaient à rendre compte de leur anticipation dans le travail de groupe à faire : des contenus et des connaissances qu'ils pensaient mobiliser, de leur organisation prévue, des difficultés auxquelles ils s'attendaient à être confrontés...

À mi-parcours, fin mars, les entretiens ont visé à analyser la mise en œuvre du cahier de charges : l'évolution du travail par rapport aux objectifs de départ, les difficultés réellement vécues par rapport à celles auxquelles ils s'attendaient, les stratégies mises en œuvre dans la gestion du projet pour être dans les délais de remise, mais aussi pour gagner la compétition et enfin, l'état d'esprit dans lequel les étudiants se trouvaient en ce moment du projet. Les derniers entretiens avec les étudiants ont eu lieu après la soutenance des projets en début du mois de juin. Ils visaient à établir un bilan : satisfaction ou pas du travail fourni, analyse des stratégies mises en œuvre dans la programmation des robots et des motivations de leurs choix, difficultés réellement vécues, critiques sur ces projets : adéquation sujet de projet/niveau, conditions de leur organisation et leurs éventuelles participations à la compétition robotique qui était prévue en juin de la même année scolaire et stratégies pour la gagner. Si les trois groupes d'étudiants ont été observés, un n'a participé qu'au seul entretien d'après la soutenance. Après la soutenance, un entretien semi directif et en groupe a aussi été mené auprès des enseignants, encadrants de ces projets. Il visait à donner plus d'éclairage sur les questions de recherche de cette contribution.

Présentation des résultats

Quelques stratégies utilisées par les étudiants

Le balayage pour cartographier le terrain de jeu est une stratégie qui a été mise en œuvre en deux temps. D'abord, le robot étant dans sa position initiale, cela lui permet de prendre connaissance de tout ce qui se trouve sur terrain : palets, lignes et leurs couleurs, robot adverse et tout autre objet éventuel pouvant constituer un obstacle. Cela est conditionné par la connaissance qu'a le robot de sa position actuelle. Au moyen d'un balayage horizontal, il s'approprie tout ce qui s'y trouve en se localisant d'abord lui-même. Ensuite, il utilise ces informations pour se déplacer. Une fois le palet en contact du capteur de pression, il ferme ses pinces, initialement ouvertes, pour le ramasser.

La deuxième stratégie a été de se précipiter pour ramasser d'abord le palet situé au milieu du terrain. Elle a un double avantage. D'abord, la configuration du terrain n'est pas encore changée et les balles sont dans leur position d'origine. Ensuite, elle permet de marquer le premier but, en premier avant le robot adverse, pour bénéficier d'un bonus (Règlement de la compétition, 2012) : « *Un bonus de 3 est accordé à l'équipe qui dépose la première balle dans l'en but adverse* ». Pour accéder à ce palet, le déplacement suivant les lignes nécessitait beaucoup de temps. Il a donc été préférable que le robot, à partir de sa position initiale, fasse une rotation d'un certain angle, initialement bien déterminé après plusieurs essais à l'avance pour se déplacer en diagonale.

Prendre le chemin le plus court pour atteindre le palet cible est une autre stratégie utilisée par tous les groupes. Une fois les palets identifiés sur terrain, des distances entre eux et le robot étaient calculées puis comparées. C'est la plus courte distance qui était parcourue par le robot pour se diriger vers le palet le plus proche. Cela lui permet d'atteindre la zone d'en-but rapidement et ainsi de marquer le but. Après ce premier parcours vers la zone d'en-but adverse, des calculs de la plus courte distance sont de nouveau faits. Le palet le plus proche du robot, à partir de la zone d'en but adverse est ciblé. Dans son parcours, il est toujours question de faire stratégiquement un bon choix sur la façon dont le robot tourne. Des paramètres tels que la distance à parcourir, l'angle sous lequel il doit tourner, le côté vers lequel il doit aller, la vitesse à laquelle il va se déplacer... sont autant d'éléments très importants pris en compte dans la programmation des robots pour les rendre plus performants. Pour se diriger vers ce nouveau palet cible, la stratégie du groupe « L2_MS_2H » est unique : pour ne pas bousculer les autres palets, des chemins extérieurs du terrain sont privilégiés. Pour la programmation, ce groupe prend le terrain pour une matrice où, les palets sont considérés comme ses éléments se trouvant à l'intersection des lignes et colonne de cette matrice.

On observe aussi la technique qui consiste à tenter à déstabiliser l'adversaire : appréhender ses stratégies pour mieux les neutraliser. Elle consiste à empêcher le robot adverse, par tous les moyens possibles, de marquer rapidement des points : « s'introduire directement dans le camp adverse pour voler les palets qu'il voudrait prendre », « sécuriser chez nous en dégageant notre terrain ».... Elle a été utilisée par un seul groupe, L2_MS_2H, qui affirme avoir eu des difficultés dans sa mise en œuvre : elle nécessite de « se mettre dans la peau de l'ennemi » et « un certain travail, un effort supplémentaire ». Tentée par le même groupe, la stratégie de boussole a été utilisée pour simuler une sorte de géolocalisation. Des risques estimés de son utilisation, notamment les risques de sa désynchronisation de la réalité au moindre choc sur terrain, sans moyen de le remettre à jour, ont fait qu'elle ne soit pas mise en œuvre.

On a aussi relevé un déplacement en zigzag, permettant de retrouver les palets bousculés lors des mouvements des robots sur le terrain, qui se retrouveraient décalés de leur position

initiale, ainsi qu'un déplacement en spirale. Mise en œuvre par le seul groupe « L2_DJ_5H », ce choix a été motivé par l'idée du groupe selon laquelle, dans son mouvement vers la zone d'en but, un robot peut, de façon involontaire, bousculer certaines balles. Ces dernières peuvent par conséquent, être plus ou moins légèrement écartées de leurs positions initiales sur terrain. Par un déplacement en spirale, le groupe imaginait permettre au robot d'accéder aux balles qui se seraient décalées de leur position initiale. Cette stratégie n'est conçue que pour être mise en œuvre après avoir fait le premier mouvement vers la zone d'en but adverse.

Après avoir ramassé la balle, les robots devaient se déplacer jusqu'à la zone d'en-but afin de déposer le palet. Le robot était programmé pour suivre les couleurs des lignes sur terrain. La zone d'en-but, de couleur blanche, était le seul indicateur de cette zone qu'il fallait à tout prix identifier. La stratégie utilisée était de détecter la couleur blanche : une fois la couleur blanche détectée, le robot était sûr de se retrouver dans la zone favorable et ainsi pouvait déposer son palet par simple ouverture de ses pinces. La synthèse des précédentes stratégies est présentée dans le tableau 1 suivant.

Tableau 1 : Variations des stratégies utilisées selon les robots, les tâches et les groupes

Tâche	Stratégie utilisée	Groupe concerné
Cartographier le terrain de jeu	Balayage avec la caméra	Tous les groupes
Géolocaliser les palets sur terrain	Calcul de l'angle et de la distance (LEGO, palet)	Tous les groupes
	Considération des palets comme éléments d'une matrice	L2_DJ_5H
	Création d'un système de zones : découpage du terrain en plusieurs zones suivant les lignes, dans lesquelles le robot se situerait : échec de délimitation	L2_DJ_4H
Gagner le bonus	Cible du palet central + déplacement en diagonale	Tous les groupes
Rechercher un plus court chemin à suivre	Calcul de la distance minimale	Tous les groupes
Déplacement vers le palet choisi	Déplacements en spirale avec balayage en suivant une ligne sélectionnée par sa couleur	Tous les groupes
	Déplacements en zigzag avec balayage en suivant une ligne sélectionnée par sa couleur	Tous les groupes
	Déplacements guidés par une boussole : échec d'utilisation	L2_MS_2H
Ramasser le palet	Le palet, au contact du capteur de pression, des pinces initialement ouvertes se ferment	Tous les groupes
Empêcher l'adversaire de marquer des points	Déstabilisation du robot adverse	L2_MS_2H
Localiser de la zone d'en-but	Détection de la couleur (blanche) de la zone	Tous les groupes

Des connaissances informatiques utilisées

Les connaissances informatiques déjà acquises utilisées par les étudiants dans la programmation du robot LEGO MINDSTORM NXT sont diverses. Le tableau 2 suivant les

présente de manière synthétique.

Tableau 2 : Notions informatiques intervenues dans la programmation de LEGO

Niveau de connaissances	Type de connaissances	Groupe concerné
Notions de base	Variables, instructions, affectation, algorithme, condition, alternatives, boucles, itérations...	Tous les groupes
Notions avancées	Fonctions, structures, pointeurs	Tous les groupes

À côté des connaissances informatiques déjà utilisées, beaucoup d'autres encore sont acquises. L'objectif des projets n'était pas l'apprentissage d'un langage, aucun langage ou technologie n'était obligé. Des choix libres sont fonction de la motivation à l'apprentissage, comme le témoigne l'encadrant E1 : « au niveau des langages, pas une limitation et on ne cadre pas sur les langages utilisés ou les technologies utilisées. (...) il y en a qui manipulent des langages qu'ils n'avaient jamais manipulés avant ». Le langage C avait été acquis en L2 au premier semestre de la même année : il était familier aux étudiants. D'autres notions acquises sont essentiellement plus techniques notamment celles régissant le fonctionnement du LEGO. En pratique, deux langages nouveaux, ont été utilisés : NXC et BrixCC, proches du langage C. Les principales connaissances informatiques construites au cours de ces activités sont présentées dans le tableau 3 suivant.

Tableau 3 : Types de connaissances informatiques acquises en fonction du robot programmé

Domaine des connaissances	Types de connaissances	Nom du groupe
Langages	NXC	L2_DJ_4H
	BrixCC	L2_MS_2H ; L2_DJ_5H
Librairies et logiciels	Firmware, Simulateur	Tous les groupes
Fonctions traitant des mouvements	OnFawdReg ()	Tous les groupes
Recherche informationnelle	Navigation sur Internet : usage des moteurs de recherche et des mots clés	Tous les groupes
Documentation en anglais	Langue anglaise	Tous les groupes

Des connaissances mathématiques élémentaires mises en oeuvre

Dans la programmation de LEGO, les étudiants semblent n'être pas confrontés aux difficultés engendrées par des connaissances mathématiques à mobiliser. Seules des connaissances élémentaires, vues dans la scolarité obligatoire, sont intervenues. Interrogés sur les notions mathématiques utilisées, cet étudiant, dans sa réponse, semble ne pas les considérer comme des mathématiques : « Pas de connaissances (mathématiques) mais des souvenirs du collège et lycée... ». L'encadrant E1 justifie cette absence de construction de nouvelles connaissances mathématiques par le fait que, dans ce contexte de la programmation, « les mathématiques sont utilisées comme *outil*, mais pas comme un objet d'apprentissage, donc pas comme un objet d'étude en tant que tel. ». La synthèse des notions mathématiques utilisées est présentée dans le tableau suivant.

Tableau 4 : Notions et domaines mathématiques intervenant dans la programmation des robots

Domaine	Notions	Groupe concerné
Géométrie	Repère, double dimension, vecteur, coordonnée, rotation, distance	L2_DJ_5H
Trigonométrie	Angle, fonctions circulaires : sinus, cosinus...	Tous les groupes
Algèbre	Matrices, homographie	L2_MS_2S

Si en mathématiques des connaissances construites sont rares, de nouvelles sont acquises dans d'autres domaines : recherche d'informations sur Internet, confrontation à une documentation en anglais, gestion des relations humaines, utilisation des logiciels pour communiquer à distance au moment du travail en synchrone, notions de physique, de mécanique, etc.

Quelques apports des projets de programmation

Une mise en œuvre de la pluridisciplinarité

Les enseignants-encadrants soulignent six principaux types de compétences développés par cette approche par projet : l'aspect « passage à l'échelle » et l'aspect « intégrateur », la reconnaissance de ses limites des étudiants, l'acquisition de nouveaux langages de programmation, l'apprentissage de la gestion des relations humaines du fait que cet apprentissage a lieu en groupe et l'apprentissage de l'anglais imposé par une documentation de plus en plus disponible dans cette langue. Le passage à l'échelle, permis par les projets, est vu comme une acquisition fondamentale chez les étudiants. Selon eux, cette forme d'apprentissage n'est nulle part ailleurs offerte que dans les projets : « Un point important est qu'ils apprennent à passer à l'échelle. D'habitude en TP ou TD, ils font des exercices de programmation ou de gestion de projet mais qui vont être assez scolaire. Par exemple quand ils ont un programme à faire en TP, ils font un programme de 30 à 40 lignes. Dans le cadre d'un projet, ils doivent gérer leur temps, et à la fin, le travail attendu peut faire plusieurs milliers de lignes de code. Et donc, l'approche qu'on a du programme n'est pas la même. Ça, je pense que le passage à l'échelle est un truc qui est assez important qu'ils ne peuvent pas apprendre ailleurs que dans les projets. ». Ce passage à l'échelle permet non seulement l'entrée en détail dans l'utilisation des cours d'informatique mais aussi l'implémentation d'un code de grande taille.

L'aspect intégrateur est le second apport de cette approche. En effet, le contexte de projet permet une mobilisation et réutilisation des connaissances déjà acquises et ce, dans beaucoup de cours, mais aussi un approfondissement de l'informatique. Selon l'encadrant E1, rares sont les cours qui permettent une telle mobilisation des connaissances : « ils [étudiants] doivent taper dans plusieurs cours et ils doivent essayer d'imbriquer. Ils doivent faire la construction des connaissances à partir des briques de base qu'on leur a données. Ça, il y a peu de cours qui permettent de le faire. Généralement, on est très pointu, on va aller dans les détails du cours d'informatique pour présenter un certain nombre de notions, mais on ne va jamais s'abstraire pour dire, bon voilà, on connaît toutes ces notions, et alors comment on les utilise. C'est aussi quelque chose qui est appris aux étudiants. ». Avec cet aspect, plusieurs connaissances et compétences issues de plusieurs cours (informatique, mathématique, physique...) sont mobilisées pour construire d'autres connaissances lors de la résolution d'un problème donné.

Une formation à l'autoformation

Les projets se distinguent des autres contextes d'enseignement et/ou d'apprentissage. Alors qu'habituellement, les enseignants cherchent pour les étudiants, l'auto-apprentissage est un apport essentiel de l'approche par projet. L'enseignant-encadrant E1 justifie cet auto-apprentissage par le fait qu'au cours du projet la totalité ce travail revient désormais aux étudiants qui construisent eux mêmes leurs connaissances : ils sont mis dans des contextes où eux-mêmes doivent tout faire pour accéder à l'information : « Voilà, on donne la matière, et puis les étudiants vont les consulter. Mais là, on est plus dans une démarche où c'est à l'étudiant qui, à partir d'un problème, c'est vraiment une démarche pour les problèmes, et il va chercher l'information. ». Les enseignants trouvent cette démarche d'autoformation « extrêmement importante » pour être inculquée aux étudiants. Trois raisons sont données pour justifier sa pertinence.

D'abord, l'autoformation en informatique fonctionne assez bien étant donné que la plupart des logiciels sont documentés (E1) : ils connaissent le projet un mois et demi avant le début avant le début du projet, la plupart commencent, tout seuls, par faire les tutoriaux associés aux technologies qu'ils vont devoir manipuler au cours du projet, et tout seul peut faire des tutoriaux associés aux technologies qui seront validées par l'encadrant. La deuxième raison donnée est l'évolution rapide des technologies qui nécessite d'être à jour (E1) : « dans la vie de tous les jours d'un informaticien, étant donné que les technologies évoluent très vite, même si on arrive à former quelqu'un d'extrêmement très pointu et qui connaît toutes les technologies, quand il sort de l'université, au bout de cinq ans, les bases sont là mais les technologies seront complètement changées ». Enfin, une dernière raison est de l'amener à faire un esprit critique des productions des autres et les améliorer éventuellement (E2) : « ils font beaucoup de tutoriaux et ils improvisent aussi quelques-uns (...). Quelques fois, ils constatent que les tutoriaux qui existent ne correspondent pas à ce qu'ils attendent ou sont en anglais, et de leurs côtés, des fois pour leur propre équipe, ils font de petits tutoriaux qu'ils correspondent plus à la situation à laquelle ils sont ».

Des interactions au service de la construction des connaissances

Ce travail en projets de programmation des robots a occasionné diverses interactions faisant intervenir plusieurs acteurs : entre les membres d'un groupe ou entre les groupes ou même entre un groupe et son encadrant. Au sein d'un groupe, ces interactions ont souvent lieu lors de mises en commun de leurs productions soit individuelles, soit issues de petits groupes formés à cet effet. À ce moment des échanges voire des discussions avaient lieu entre deux étudiants, trois ou tous les membres du groupe. À l'issue de ces interactions au sein d'un groupe, certaines productions étaient mises de côté pour ne garder que celles jugées meilleures qu'il fallait compléter pour les améliorer ensemble.

Si de tels des échanges pour la construction des connaissances avaient lieu et étaient fréquents au sein d'un groupe sur le même projet, d'autres échanges pouvaient avoir lieu entre des groupes différents mais travaillant sur un même projet : les échanges consistaient en des explications de ce qu'ils n'ont pas pu comprendre ou des stratégies à utiliser, des utilisations des logiciels non maîtrisés, etc. Bien que ces échanges soient bénéfiques, ils n'ont pas été toujours possibles au sein de tous les groupes en projets qui travaillaient sur le même sujet. Le groupe, L2_DJ_5H, « content de réussir ensemble avec les autres », déplore un refus de collaboration de certains groupes ciblés qui travaillaient sur un même projet qu'eux : « certains sont réfractaires à l'échange, surtout un groupe particulier ! ». Ce groupe, tout en critiquant ceux de sa classe qui ne parvenaient pas à se parler, rendait hommage aux étudiants de L3 dans leur façon de travailler et de collaborer avec les autres : « Les L3 nous aident, nous, alors qu'on était sur des trucs différents. (...). On parlait avec les L3 mais avec les L2, non ! ».

Quant aux questions des étudiants, elles sont selon E2, des questions avec enjeu, donc moins naïves ; des questions allant dans le sens de débloquer une situation afin d'aller plus loin, et enfin, des questions qui cherchent à savoir quelle question les bons groupes ne se sont pas posée pour débloquer la situation, plutôt que de chercher la réponse à leur question auprès de lui. Face à ces questions, les interventions de l'encadrant sont restées essentiellement orales, suscitant l'initiative des étudiants. L'enseignant E1 affirme donner rarement une réponse directe à la question posée : « Il est évident qu'on ne peut pas connaître tous les détails, toutes les astuces de toutes les technologies. Par contre, on peut les pousser à adopter justement une démarche de recherche de l'information pour les amener à s'interroger. Par exemple moi, je donne jamais ou très rarement la réponse à la question ».

Discussion

De la conception à l'implémentation d'une tâche : un espace pour la « pensée algorithmique »

Face au problème donné, les étudiants devaient faire preuve d'une mobilisation d'une pensée algorithmique en concevant une méthode pour sa résolution : la déconstruction du problème. Non seulement ils devaient imaginer, de manière abstraite ce qui se passe et comment ça se passe, comment le robot va se déplacer, où il va commencer, les fonctions à prévoir, comment les nommer et les programmer... Aussi, ils devaient arriver à anticiper le fonctionnement de cet algorithme lors de son exécution, notamment comment les fonctions seront appelées, dans quel ordre lors d'une exécution dynamique du programme, sur l'ordinateur que sur le robot. Cette mobilisation de la pensée au cours de l'activité de programmation a été soulignée dans beaucoup de recherches.

L'un des buts de l'informatique étant le développement de la « pensée algorithmique par la formulation d'une méthode de résolution de problèmes » (Chaguiboff, 1985), les résultats révèlent que les étudiants conçoivent les méthodes à utiliser mais ne savent pas forcément faire le choix de la méthode la plus efficace : certaines stratégies abandonnées laissent voir que c'est la facilité qui est privilégiée plutôt que la rentabilité ou la qualité des apprentissages permis. La littérature de recherche donne deux difficultés essentielles de la programmation (Duchâteau, 1993) : celle liée à conception d'une marche à suivre et celle liée aux traits spécifiques de l'exécutant. Cette dernière difficulté est plus vécue par les étudiants. Selon lui, « on ne peut être un concepteur que si, en même temps, on est capable de se mettre dans la peau de l'exécutant » pour anticiper son exécution, prévoir les possibilités. Ainsi, un pied est laissé dans le monde de la tâche à faire exécuter et l'autre dans l'univers de l'exécutant et de ses possibilités. En recommandant que l'apprenant débutant en programmation affronte ces difficultés, il confirme ainsi les propos de Jean Chaguiboff pour qui la programmation est plus aisée pour le débutant qui dispose de représentations préalables de l'exécutant, d'où la nécessité d'apprendre son fonctionnement.

Les projets robotiques, des contextes adaptés à l'apprentissage ?

Tout comme un ordinateur peut être utilisé dans toutes les disciplines et permettre l'apprentissage de divers savoirs spécifiques aux disciplines, la robotique pédagogique ne déroge pas à la règle de pluridisciplinarité. Duchâteau (1993, p.1) évoque un éclatement des frontières disciplinaires comme caractéristique fondamentale commune à l'informatique et à la robotique pédagogique. Les exigences, les réflexions... suscitées par la programmation des robots semblent permettre aux apprenants d'aller au-delà des limites disciplinaires pour interpréter les comportements de l'exécutant robot.

Un autre aspect important est en rapport avec des interactions diverses vécues en situations d'apprentissage. Les tâches sont partagées entre les membres du groupe. Chacun a des occasions d'interagir avec son ordinateur et le robot à programmer. Après, la réponse est

validée après consensus du groupe. Ces interactions qui pouvaient aller au-delà d'un seul groupe selon la difficulté de tâche à faire, offre un contexte socioconstructiviste d'apprentissage, où les forts intellectuellement complètent les faibles. Cette complémentarité entre étudiants leur a permis d'apprendre plus et d'aller plus loin dans l'approfondissement et la compréhension de leurs apprentissages en confrontant leur point de vue. L'enseignant se trouve dans les nouvelles conditions de travail (Janiszek et al, 2011b, p.238-239) : il n'est qu'un guide dans la plupart des cas et n'intervient pas directement dans le travail des étudiants dont il ne maîtrise pas la démarche adoptée. Pour tenir compte des besoins des étudiants, il agit en médiateur. Il gère donc l'incertitude et se contente de donner de nouvelles orientations pour débloquer ou pour permettre d'aller plus loin.

Deux constats, constituant à notre sens les avantages de cette approche par projet, peuvent être soulignés. D'abord, ce sont les stratégies communes discutées qui ont été les plus réussies à mettre en œuvre. Ensuite, un groupe qui ne veut pas communiquer avec les autres était critiqué et considéré comme marginal, ce qui témoigne encore de l'importance accordée à la collaboration lors du travail en groupe chez les étudiants. Nos résultats révèlent aussi que c'est lorsqu'il y a des difficultés qu'il y a de fortes interactions entre apprenants, interactions qui sortaient du groupe pour atteindre d'autres groupes, ce qui favorise l'apprentissage. Ils finissaient généralement par arriver à surmonter la difficulté, même si cela pouvait prendre beaucoup de temps. Privilégiant la complémentarité, le recours à l'enseignant avait lieu à la limite, et permettait d'avoir des orientations, mais sans toutefois travailler à leur place. Ces pratiques des étudiants, généralement des essais et erreurs répétitifs qui occasionnaient des va-et-vient des étudiants de l'ordinateur au robot et vice versa au moment de tests des codes, sont, d'après nous d'excellentes occasions d'apprentissages chez les étudiants. Ces apprentissages au cours de périodes de tâtonnements et de répétitions ont été déjà confirmés par des recherches (Lagrange, 1993, p.119).

Perspectives

L'approche par projet de la programmation des robots semble une approche permettant non seulement la construction de nouveaux savoirs mais aussi l'utilisation de ceux déjà acquis. Par son caractère pluridisciplinaire, les connaissances tant utilisées que construites se retrouvent dans plusieurs domaines. La maîtrise d'une technologie nouvelle telle qu'un robot exige plus qu'une simple connaissance des composantes du hardware, de la syntaxe et la sémantique de programmation, mais aussi une connaissance préalable des principes de son fonctionnement (Gaudiello & Zibetti, 2013). Malgré les compétences reconnues, cela leur manquait : le robot était une boîte noire pour les étudiants. Notre hypothèse est que des séances d'initiation et de familiarisation à la structure et au fonctionnement du robot avant les projets leur permettraient d'acquérir des images mentales nécessaires et suffisantes de cette boîte noire, pour aller plus loin dans leur apprentissage de la programmation. Les étudiants de licence en informatique ont pas mal de cours orientés vers l'apprentissage de la programmation (Delozanne et al., 2011). Il nous semble que ces étudiants qui venaient de passer plus d'une année entière à l'apprentissage de la programmation, avaient déjà acquis un bon niveau de pratiques : ils n'ont pas eu de difficultés particulières dans la programmation classique. N'est-il pas intéressant de voir si les apprenants (élèves ou étudiants) débutants qui se confrontent à la programmation pour la première année vivent les mêmes difficultés que celle de L2 ayant été observés et interviewés dans cette recherche ?

Références

- Arnaud, P. (1999). *Des moutons et des robots : Architectures de contrôle réactive et déplacements collectifs de robots*. Presses Polytechniques et universitaires romaines.
- Baron, G. L., & Denis, B. (1993). *Regards sur la robotique pédagogique. Actes du 4e colloque international sur la robotique*

- pédagogique. Paris : INRP, Technologies nouvelles et éducation.
- Bonnel, B. (2010). *Vive la robolution*. Paris : Editions JClattès.
- Chaguiboiff, J.(1985). Informatique et apprentissages. In *Enfance*, Tome 385(1), 31-42
- Delozanne, E., Jarraud, P. et Muratet, M. (2011). Un projet Jeux sérieux pour approfondir l'apprentissage de la programmation en première année à l'université In *Baron, G-L., Bruillard, E. et Komis, V. (Eds) (2011). DIDAPRO 4-dida & STIC : Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif*. Actes de Colloque International, 24-26 octobre, Patras, Grèce, 240-249.
- Duchâteau, C.(1993). Robotique-Informatique : mêmes ébats, mêmes débats, mêmes combats, in Denis, B., & Baron, G-L. (Eds.). (1993). *Regards sur la robotique pédagogique- Actes du 4ème colloque sur la robotique pédagogique*. Paris / Liège: INRP / Université de Liège, 10–33.
- Duchâteau, C. (2002). *Images pour programmer*. CeFIS-FUNDP, Namur, 183p.
- Gaudiello, I., & Zibetti, E. (2013). La robotique éducationnelle : état des lieux et perspectives. *Psychologie Française*, 58(1), 17-40. doi : 10.1016/j.psfr.2012.09.006
- Gaudiello, I., Zibetti, E., & Pinaud, C.-A. (2012). Control heuristics for educational robots : a pilot study. In *Proceedings of 3rd International workshop teaching robotics, teaching with robotics. Integrating robotics in school curriculum, Riva del Garda, Trento, Italy* (p. 67–75). Consulté à l'adresse http://www.terecop.eu/TRTWR2012/trtwr2012_submission_09.pdf
- Janiszek, D. et al. (2011b). De l'usage de Nao (robot humanoïde) dans l'apprentissage de l'informatique. In Baron, GL, Bruillard Eric et Komis Vassilis (Eds) : *Didapro4-Dida&STIC : Sciences et technologies de l'information et de la communication en milieu éducatif. Actes du Colloque International, 24-25 octobre 2011, Université de Patras, Grèce*, p. 231-239
- Janiszek, D. et al. (2011a). Utilisation de la robotique pédagogique pour enseigner l'intelligence artificielle : une expérience d'approche par projet auprès d'étudiants en informatique http://sticf.univ-lemans.fr/num/vol2011/07r-janiszek/sticf_2011_janiszek_07rp.pdf
- Kynigos, C. (2008). Black and white perspectives to distributed control and constructionism in learning with robotics. *Workshop Proceedings of SIMPAR*.
- Komis, V., & Misirli, A. (2011). Robotique pédagogique et concepts préliminaire de la programmation à l'école maternelle : une étude de cas basée sur le jouet programmable Bee-Bot. In *Didapro4-Dida&STIC : Sciences et technologies de l'information et de la communication en milieu éducatif. Actes du Colloque International, 24-25 octobre 2011, Université de Patras, Grèce*, 271-281.
- Laborde, C., Mejias, B., & Balachef, N. (1985). Genèse du concept d'itération : une approche expérimentale. *Enfance*, 38(2-3), 223–239.
- Lagrange, J.-B. (1993). Les objets et les types dans un enseignement de la programmation s'adressant à des débutants. *Actes de la troisième rencontre francophone de didactique de l'informatique Sion, du 6 au 11 juillet 1992*, 115–120.
- Nonnon, P. (2002). Robotique pédagogique et formation de base en science et technologie. *Aster*, 2002, 34« Sciences, techniques et pratiques professionnelles ». Consulté à l'adresse <http://documents.irevues.inist.fr/handle/2042/8787>
- Marchand, D. (1992). La robotique pédagogique! \cca existe. *Bulletin de l'EPI*, (65), 119–124.
- Marcel, J.-F. (2002). Approche ethnographique des pratiques enseignantes durant les temps interstitiels. *Revue Spirale*, 30, 103–120.
- Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules : Concrete-abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology and Design Education*, 19(1), 15–36.
- Paulin, M., Bourreau, E., Dartnell, C., & Krut, S. (2006). Modélisation et planification d'actions élémentaires robotiques par apprentissage de réseaux de contraintes. Deuxième Journées Francophones de Programmation par Contraintes (JFPC06), Nîmes-Ecoles des Mines d'Alès/France. Consulté 12 mars 2013, à l'adresse <http://hal.archives-ouvertes.fr/inria-00085797/>
- Règlement de la compétition Robot PRES Cup, (2012). Edition 2012.
- Roby-Brami, A., & Laffont, I. (2002). Gestes et technologie : la compensation des incapacités motrices. In *Blandine Bril et Valentine Roux (dir) : Le geste technique : Réflexions méthodologiques et anthropologiques*, érès, Ramonville Saint-Agne, 95-112.
- Tardif, J. (1997). Pour un enseignement stratégique : l'apport de la psychologie cognitive : *Les Ed. Logiques*.